



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re: Claussen et al.

Serial No.: Claussen et al.
09/409,370

Filed: 09/30/1999

Method for processing a
document object model (DOM)
tree using a tagbean

S Group Art Unit: 2176

S Examiner: Singh, R.

S Atty. Docket #: AT9-99-485

S
S
S
S
S

5

DECLARATION UNDER 37 C.F.R. § 1.131

To the Commissioner for Patents:

Christopher Shane Claussen declares as follows:

10

1. I am an applicant for the patent application entitled "Extensible Markup Language (XML) Server Pages Having Custom Document Object Model (DOM) Tags", U.S. Patent Application Serial Number 09/409,598, filed September 30, 15 1999, and an inventor of the subject matter described and claimed therein.

15

20 2. Prior to July 1, 1999, I completed and reduced to practice, in the United States of America, the invention described and claimed in the above-noted application, as evidenced by the following:

20

25 (a) I was named as an inventor on IBM Invention Disclosure Form AUS8-1999-0688, a document of which a copy is attached hereto as Exhibit "A", which describes the invention that is described and claimed in the subject application.

Page 1

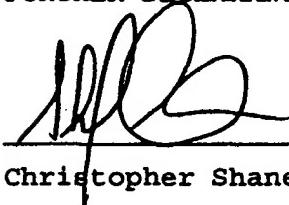
Claus Claussen et al.
09/409,370

3. Each of the dates deleted from Exhibit "A" is prior to July 1, 1999.

4. I am named on page xviii (the "Acknowledgements" 5 page) of the "JavaServer Pages™ Specification" Version 1.1 - Public Release", Sun Microsystems, Inc., 08/18/1999, as having contributed to the JSP 1.1. specification.

5. I further declare that all statements made herein of 10 my own knowledge and all statements made on information and belief are believed to be true; and further that these statements are made with the knowledge that willful and false statements and the like so made are punishable by fine or imprisonment or both under § 1001 of Title 18 of 15 United States Code and that such willful and false statements may jeopardize the validity of the above-referenced application and any patent issuing therefrom.

20 FURTHER DECLARANT SAYETH NOT.



Christopher Shane Claussen

Date:

4/8/04

25

EXHIBIT "A"

Page 3
Claus Claussen et al.
09/409,370



Disclosure AUS8-1999-0688

Created By: Ben ZumBrunnen Created On: [REDACTED]

Last Modified By: Matthew McClain Last Modified On: [REDACTED]

*** IBM Confidential ***

Required fields are marked with the asterisk (*) and must be filled in to complete the form.

Summary

Status	Under Evaluation
Processing Location	AUS
Functional Area	46 -SWS- TECHNOLOGY CENTER (W. W. Lawton)
Attorney/Patent Professional	Jeff LaBaw/Austin/IBM
IDT Team	Craig Becker/Austin/IBM
Submitted Date	[REDACTED]
Owning Division	SWSD [REDACTED]
PVT Score	To calculate a PVT score, use the 'Calculate PVT' button.

Inventors with Lotus Notes IDs

Inventors: Ben ZumBrunnen/Austin/IBM, Shane Claussen/Austin/IBM, Matthew McClain/Austin/IBM, Lin Xu/Austin/IBM

Inventor Name > denotes primary contact	Inventor Serial	Div/Dept	Manager Serial	Manager Name
ZumBrunnen, Benjamin C. (Ben)	808909	76/R37A	374443	Lawton, W.W. (Bill)
> Claussen, C.S. (Shane)	595375	76/R37A	374443	Lawton, W.W. (Bill)
McClain, Matthew D.	932145	76/R37A	374443	Lawton, W.W. (Bill)
Xu, Lin	652075	76/R37A	374443	Lawton, W.W. (Bill)

Inventors without Lotus Notes IDs

IDT Selection

IDT Team:	Craig Becker/Austin/IBM	Attorney/Patent Professional:	Jeff LaBaw/Austin/IBM
-----------	-------------------------	-------------------------------	-----------------------

Response Due to IP&L: [REDACTED]

Main Idea

*Title of disclosure (in English)

Custom DOM tags

*Idea of disclosure

1. Describe your invention, stating the problem solved (if appropriate), and indicating the advantages of using the invention.

Problem: Server-side scripting on a web page can be messy, and therefore, difficult to maintain and evolve.

2. How does the invention solve the problem or achieve an advantage,(a description of "the invention", including figures inline as appropriate)?

Solution: Custom DOM tags. Our invention is a framework and runtime providing a powerful macro language to XML/JSP. Custom DOM (Document Object

Model in eXtensible Markup Language speak) tags allow a web page author to define a simple markup tag, say, <SHOPPING_CART>, that at page-translation time, a JavaBean or XSL stylesheet converts into script code. Further, this script code is then compiled into Java code, and then into a Java servlet, yielding excellent performance servicing a client's request. Since the custom tag replaces the script code in the authored page, the page is kept clean and easy to maintain. The script code is kept separate, and therefore, need only be debugged once. Normal ASP development would force this code to remain in the page, and it would have to be debugged after every modification.

3. If the same advantage or problem has been identified by others (inside/outside IBM), how have those others solved it and does your solution differ and why is it better?

Macromedia's Dreamweaver has a tag construction mechanism, but it keeps the script code on the page. The script is hidden from the user within Dreamweaver, but is still present on the page, and may be viewed/edited from within another editor. The modifications made in another tool will not round-trip back into Dreamweaver. Since our implementation keeps the script code out of the page, we round-trip between any editor.

Microsoft has its design-time control (DTC) technology, and this is similar to MM's Dreamweaver. A DTC hides its code from the user, but still keeps it on the page. While other DTC-aware tools will hide the script code correctly, basic text-editors can still edit the code, and changes made will not be importable into the DTC-aware tools.

Other products that have some custom tag extensions are HeiTML (<http://www.heitml.com/heitml1.2/index.hei?rmenu=Introduction>), Meta-HTML (<http://www.metahtml.com/>), and_hpp (<http://debra.doc.ca/doodads/hpp.html>).

_hpp is just an HTML preprocessor.

HeiTML and Meta-HTML are very similar to one another
They are Tag Macro definition languages for HTML.
They have control flow and limited scripting: <when> <if> <not>
They have tags for Database access and form handling:
either might be a good target for an XSP namespace.
Not XML compliant: don't use end-tags

1) HeiTML

Can send tasks to the command shell.
Can redefine HTML tags.
Not XML compliant:
values in tags aren't attributes (just parameters)

2) Meta-HTML

Someone has done work on top of Meta-HTML to define multi-language (English, French, Spanish, Chinese)
macro support.
Not XML compliant:
<when <not -- Is not XML friendly

XSP/JSPx: Custom DOM Tags - What's special and unique about our implementation

Syntax of the extensible tag is either XSL or by a Java object which supports the TagBean interface. We've talked about defining TagBeans in other languages as well (JavaScript, Lisp) but we don't have that functionality yet.

The main claims are:

- 1) built on top of XML (none of the others are)
- 2) definable programatically or with XSL
- 3) Macro definition language for XSP/JSPx (JSP with multiple computer languages and HTML)
- 4) Macros can affect the output of the whole page, not just what's in between a pair of tags.

4. If the invention is implemented in a product or prototype, include technical details, purpose, disclosure details to others and the date of that implementation.

XSP (XML Server Pages) Extension to WebSphere, plan to ship to Alphaworks 7/99.

In this implementation, custom tags are registered through an XML <directive.page taglib="tag-library.xml"> tag, according to the JSP 1.0 specification. In the tag-library.xml file, the name of the custom tag is listed, as well as a JavaBean name or URL naming an XSL stylesheet. The first time the page is hit, our engine parses it into an XML Document Object Model (DOM), and we load all the required tag libraries. We then walk the DOM tree, looking for registered custom tags. Upon finding one, if it is registered as a JavaBean, we load the bean, and call a "process" method upon it, passing the custom tag's tree node. The JavaBean then examines the custom tag, and replaces it with other custom tags, script code, or HTML. The JavaBean has access to the entire DOM, and may modify it however it deems necessary. Once processing is complete, we rescan the DOM to pick up any new tag library directives and new custom tags, and continue invoking the tag bean handlers.

An interesting possibility is when the tag bean handler is registered as an XSL stylesheet. In this case, instead of simply handing the tag bean element to the handler, we load the stylesheet, and pass it and the DOM to an XSL processor. The XSL processor applies the template to each instance of a custom tag, and performs the same replacement that a JavaBean would. The advantage is that many more web authors know XML/XSL than those who know Java, making our custom tag mechanism appeal to both communities.

Once all custom tags are reduced to HTML and script code, we compile the DOM into a Java servlet to service the client request. The translation step only occurs on the first request - all subsequent requests are handled by the compiled servlet, yielding good performance.

*Critical Questions (Questions 1 - 7 must be answered)

*Question 1:

On what date was the invention workable? [REDACTED] Please format the date as MM/DD/YYYY
(Workable means i.e. when you know that your design will solve the problem).

***Question 2**

Is there any planned or actual publication or disclosure of your invention to anyone outside IBM?

Yes
 No

If yes, Enter the name of each publication or patent and the date published below.

Publication/Patent: Alphaworks release

Date Published or Issued: [REDACTED]

Are you aware of any publications, products or patents that relate to this invention?

Yes
 No

If yes, Enter the name of each publication or patent and the date published below.

Publication/Patent: JSP 1.0 Specification hints that this idea might be possible

Date Published or Issued: 05/28/1999

***Question 3**

Has the subject matter of the invention or a product incorporating the invention been sold, used internally in manufacturing, announced for sale, or included in a proposal?

Yes
 No

Is a sale, use in manufacturing, product announcement, or proposal planned?

Yes
 No

If Yes, identify the product if known and indicate the date or planned date of sale, announcements, or proposal and to whom the sale, announcement or proposal has been or will be made.

Product: WebSphere

Version/Release: 3.x

Code Name:

Date: 12/1999

To Whom: IBM internal

If more than one, use cut and paste and append as necessary in the field provided.

***Question 4**

Was the subject matter of your invention or a product incorporating your invention used in public; e.g., outside IBM or in the presence of non-IBMers?

Yes
 No

If yes, give a date. Please format the date as MM/DD/YYYY

***Question 5**

Have you ever discussed your invention with others not employed at IBM?

Yes
 No

If yes, identify individuals and date discussed. Fill in the text area with the following information, the names of the individuals, the employer, date discussed, under CDA, and CDA #.

Spoke with Eduardo Pelegri-Llopart of Sun Microsystems early in May, explaining we had a very powerful tag-extension mechanism and we wanted to be sure the upcoming JSP 1.0 spec did not prevent our ideas. We did not offer any details on our implementation, only worked to ensure JSP 1.0 would be as XML-compliant as reasonably possible.

***Question 6**

Was the invention, in any way, started or developed under a government contract or project?

Yes
 No
 Not sure

If Yes, enter the contract number

***Question 7**

Was the invention made in the course of any alliance, joint development or other contract activities?

Yes
 No
 Not Sure

If Yes, enter the following : Name of Alliance, Contractor or Joint Developer

Contract ID number

Relationship contact name

Relationship contact E-mail

Relationship contact phone	
Question 8	
Have you submitted, or are you aware of, any related disclosure submission?	
If Yes, please provide the title and docket or disclosure number below: We have several invention disclosures under development related to our implementation of XSP.	
Question 9	
What type of companies do you expect to compete with inventions of this type? Check all that apply.	
<input type="checkbox"/> Manufacturers of enterprise servers <input type="checkbox"/> Manufacturers of entry servers <input type="checkbox"/> Manufacturers of workstations <input type="checkbox"/> Manufacturers of PC's <input type="checkbox"/> Non-computer manufacturers <input type="checkbox"/> Developers of operating systems <input checked="" type="checkbox"/> Developers of networking software <input checked="" type="checkbox"/> Developers of application software <input checked="" type="checkbox"/> Integrated solution providers <input checked="" type="checkbox"/> Service providers <input type="checkbox"/> Other (Please specify below)	

Patent Value Tool (Optional - this may be used by the inventor and attorney to assist with the evaluation of your invention.)

(The Patent Value tool can be used by you or the evaluation team to determine the potential licensing value of your invention.)

The Patent Value Tool has not yet been used to calculate a score.

Post Disclosure Text & Drawings

Enter any additional information relating to this disclosure below:

(Form Revised 12/17/97)

BEST AVAILABLE COPY